Received: 4 October 2019 | Revised: 29 July 2020 | Accepted: 21 August 2020

DOI: 10.1002/nme.6529

SPECIAL ISSUE ARTICLE

WILEY

rp-adaptation for compressible flows

Julian Marcon[1] | **Giacomo Castiglioni**[1] | **David Moxey**[2] | **Spencer J. Sherwin**[1] | **Joaquim Peiró**[1]

[1]Department of Aeronautics, Imperial College London, London, UK

[2]College of Engineering, Mathematics & Physical Sciences, University of Exeter, Exeter, UK

Correspondence
Joaquim Peiró, South Kensington Campus, London SW7 2AZ, U.K.
Email: j.peiro@imperial.ac.uk

Funding information
Engineering and Physical Sciences Research Council, Grant/Award Number: EP/R029423/1; European Comission Horizon 2020 Marie Skłodowska-Curie Actions, Grant/Award Number: 675008

Abstract

We present a novel *rp*-adaptation strategy for high-fidelity simulations of compressible inviscid flows with shocks. The mesh resolution in regions of flow discontinuities is increased by using a variational optimizer to *r*-adapt the mesh and cluster degrees of freedom there. In regions of smooth flow, we locally increase or decrease the local resolution through increasing or decreasing the polynomial order of the elements, respectively. This dual approach allows us to take advantage of the strengths of both methods for best computational performance, thereby reducing the overall cost of the simulation. The adaptation workflow uses a sensor for both discontinuities and smooth regions that is cheap to calculate, but the framework is general and could be used in conjunction with other feature-based sensors or error estimators. We demonstrate this proof-of-concept using two geometries in transonic and supersonic flow regimes. The method has been implemented in the open-source spectral/*hp* element framework *Nektar++*, and adaptivity is performed by its dedicated high-order mesh generation tool *NekMesh*. The results show that the proposed *rp*-adaptation methodology is a reasonably cost-effective way of improving simulation accuracy.

KEYWORDS

adaptivity, compressible flow, discontinuous Galerkin, error estimation, Euler flow, fluids

1 | INTRODUCTION

The accurate and high-fidelity simulation of high-speed compressible flows is, at present, a problem of significant interest to the aeronautics community, particularly in relation to civil and military flights in which such conditions are routinely encountered. The complex and interdependent fluid phenomena found in the high-speed flow regime pose a difficult challenge for numerical modeling, with contrastingly different behaviours between regions of smooth flow, boundary layers near solid walls where large velocity gradients are present, and the interaction with shock waves or shear layers where the fluid properties change sharply in a discontinuous manner.

Abbreviations: BC: boundary conditions; CAD: computer-aided design; CFD: computational fluid dynamics; CPU: central processing unit; DG: discontinuous Galerkin; DOF: degrees of freedom; HLLC: Harten–Lax–van Leer-contact; HPC: high-performance computing; IC: initial conditions; I/O: input-output; LDG: local discontinuous Galerkin; NACA: National Advisory Committee for Aeronautics; PDE: partial differential equations

The use of high-order spectral/*hp* element methods in the simulation of compressible fluid dynamics is now becoming increasingly common for high-fidelity large-eddy simulations and direct numerical simulations of realistic aeronautical configurations.[1] As in traditional low-order methods, the domain of interest is partitioned into finite elements; however, these elements are also equipped with high-order polynomial expansions, as opposed to standard linear shape functions. This yields several advantages in terms of computational performance, as well as enhanced numerical resolution as the polynomial order p is increased. However, in the presence of shocks and other discontinuities, the latter advantage will not be realized, and can lead to significant issues in terms of stability and accuracy in the resolution of shocks.

A common approach used in the resolution of discontinuous features is to refine these regions in an adaptive manner, so that the mesh resolution around the features is increased. In broad terms, the error of a computed solution which is sufficiently smooth can be roughly expressed as $e \approx kh^p$, where k is a constant related to the measure of the solution regularity, h is the mesh size, and p is the polynomial order. Mesh adaptation is concerned with achieving increased resolution by either locally reducing the mesh size, h, or locally increasing the polynomial order, p. Due to its higher convergence rates, p-adaptation is typically preferred over h-adaptation for smooth flow regions,[2-4] whereas the opposite is true where flow discontinuities exist. The reason for the latter—h-adaptation being preferred for flow discontinuities—is that the representation of shocks by high-order discretizations leads to numerical oscillations that must be smoothed out by the addition of high-order dissipation terms. This effectively means that the high-order degrees of freedom (DOF) are wasted in the vicinity of shocks.

To address these issues, we present a proof-of-concept strategy based on *rp*-adaptation to best take advantage of h-type, through r-adaptation, and p-type local resolution modifications. For the r-adaptation procedure, a variational optimizer is used to deform the mesh.[5] By targeting a small element size in regions of shocks, the optimizer deforms the mesh and clusters nodes in said regions. By effectively redistributing DOF, h-type refinement is obtained at flow discontinuities. We then apply p-adaptation[4] to this adapted mesh to better resolve regions of smooth flow. Throughout this work, we focus on the simulation of inviscid flows and on the challenge of efficiently modeling smooth flows with embedded discontinuities.

The success of an adaptation procedure depends on the use of reliable error indicators. Different types of error indicators have been studied over the years, each with their own pros and cons. In a first category, we identify indicators based on flow or solution features, such as boundary layer, multiphase interfaces, and vortices.[6] These can be costly to evaluate and are often not robust. Another type of indicator looks at the discretization error, namely the difference between the exact and the discrete solution. The exact solution is typically unknown though and a practical workaround is to compare the solution at two different levels of accuracy, for example, the solution and its projection onto a coarser mesh[7] or onto a lower polynomial space.[8] These indicators are typically cheaper to compute but only highlight regions of high local error, even convected error. Finally, goal-oriented indicators provide sensitivities of a target quantity of interest to local mesh changes.[9] These are often based on an adjoint solution,[10] which is expensive to obtain, but they give an accurate indicator of solution error since they incorporate the physics of the problem through the computed adjoint sensitivities. However, for the problems we consider here, where shocks are a dominant feature of the flow physics, an adjoint-based error indicator may not yield the desired increase in accuracy, as shown by Ekelschot et al.[4] For a more complete review of existing error indicators for high-order computational fluid dynamics (CFD) solutions, we refer the reader to the work of Naddei et al.[11]

Here we use a discontinuity sensor[8] that is easily computed as it essentially looks at the energy of the higher modes to determine the level of resolution of the solution.

Though originally intended as a *shock sensor*, the sensor is in fact a resolution indicator based on the decay of modal energy. This characteristic makes it applicable to regions of smooth flows as well as regions containing shocks. This kind of indicator has been successfully used in various contexts, including incompressible h-adaptive spectral element solvers,[12] smooth compressible finite difference solvers,[13] and incompressible[14] and smooth compressible[11,14] p-adaptive spectral element solvers.

Because of its versatility, the purposes of this sensor in the current work are threefold: first, it adds artificial viscosity to the governing equations, based on values of the sensor, to stabilize the solution in the presence of shocks; second, it identifies regions of flow discontinuities based on values of the sensor, as used for the artificial viscosity, to drive r-adaptation; and third, it locally increases or decreases the local polynomial approximation based on the values of the sensor.

We present the proposed proof-of-concept methodology as follows. Section 2 introduces the governing equations in continuous and discrete forms. Section 3 describes the spectral/*hp* discontinuous Galerkin (DG) discretization used in *Nektar++*, with Section 3.1 covering the formulation of the discontinuity sensor and the artificial viscosity. Section 4 recalls previous work on variational r-adaptation.[5] Section 5 summarizes the p-adaptation strategy.[4] Section 6 describes the novel dual *rp*-adaptation workflow. Finally, we present two numerical examples in Section 7: a transonic flow past a

NACA 0012 aerofoil at a free-stream Mach number of 0.8 and an angle of incidence of 1.25°, and a supersonic flow at a free-stream Mach number of 3 past an engine intake that exhibits a complex shock pattern in its diffuser.

2 | GOVERNING EQUATIONS

The Euler equations of inviscid compressible flow are written, in a two-dimensional Cartesian frame of reference with coordinates $\boldsymbol{x}=(x_1, x_2)$ within a domain Ω with boundary Γ, as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F} = \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot [\ \mathbf{F}_c(\mathbf{u}) + \mathbf{F}_d(\mathbf{u}, \nabla \mathbf{u})\ ] = \mathbf{0}. \tag{1}$$

Here $\mathbf{u} = [\rho, \rho v_1, \rho v_2, \rho E]^t$ is the vector of conserved variables, where ρ is the density, the Cartesian components of the velocity are $\boldsymbol{v}=(v_1, v_2)$, and E is the total energy. The terms $\mathbf{F}_c$ and $\mathbf{F}_d$ denote the convective and dissipative fluxes, respectively. A dissipative flux is required to stabilize the solution in the presence of shocks which is chosen to be of the form

$$\mathbf{F}_d = -\mu_a(\mathbf{u})\nabla \mathbf{u}, \tag{2}$$

where μ_a is an artificial viscosity coefficient that will be discussed in detail in Section 3.1. The components of the convective flux, $\mathbf{F}_c = (\mathbf{f}_1, \mathbf{f}_2)$, are given by

$$\mathbf{f}_1 = \begin{Bmatrix} \rho v_1 \\ P + \rho v_1^2 \\ \rho v_1 v_2 \\ \rho v_1 H \end{Bmatrix}, \quad \mathbf{f}_2 = \begin{Bmatrix} \rho v_2 \\ \rho v_1 v_2 \\ P + \rho v_2^2 \\ \rho v_2 H \end{Bmatrix},$$

where H is the total enthalpy and P is the pressure. The total enthalpy is defined as

$$H = E + \frac{P}{\rho} \tag{4}$$

and, to close the system, the pressure for a perfect gas is given by

$$P = (\gamma - 1)\rho \left(E - \frac{v_1^2 + v_2^2}{2} \right), \tag{5}$$

where γ is the ratio of specific heats and its value for air is $\gamma = 1.4$.

The setting of the problem is completed through a suitable choice of initial and boundary conditions (IC/BC). Given that only steady-state problems are of interest here, we start the simulation with a uniform flow at the given free-stream Mach number and flow incidence. Solid walls are modeled through the no-flow BC, $\boldsymbol{v}\cdot\boldsymbol{n}=0$, where $\boldsymbol{n}$ denotes the wall outer normal. Far-field boundaries are weakly imposed through the normal boundary fluxes by specifying free-stream BC, $\mathbf{u} = \mathbf{u}_\infty$, outside the boundary and evaluating the normal fluxes through a Riemann solver that accounts for the propagation of information across the boundary.

3 | DISCONTINUOUS GALERKIN DISCRETIZATION

To obtain a discrete solution of the governing equations (1) via a high-order spectral/*hp* DG discretization, we assume that the computational domain, Ω, is subdivided into N_{el} nonoverlapping elements, so that $\Omega = \cup_{e=1}^{N_{el}} \Omega^e$ and $\Omega^{e_1} \cap \Omega^{e_2} = \emptyset$ for $e_1, e_2 = 1, \ldots, N_{el}$ and $e_1 \neq e_2$. We adopt a mixed formulation similar to[15] and rewrite the system (1) as

$$\mathbf{g} - \nabla \mathbf{u} = \mathbf{0}, \tag{6}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot [\ \mathbf{F}_c(\mathbf{u}) + \mathbf{F}_d(\mathbf{u}, \mathbf{g})\ ] = \mathbf{0}. \tag{7}$$

We seek a discrete approximation within an element, Ω^e, of the form

$$\mathbf{u}(\boldsymbol{x}, t) \approx \mathbf{u}_h^e(\boldsymbol{x}, t) = \sum_{i=1}^{N_{el}} \mathbf{u}_i^e(t) w_i^e(\boldsymbol{x}); \quad \boldsymbol{x} \in \Omega^e, \tag{8}$$

where $w_i^e(\boldsymbol{x}); i = 1, \ldots, N_{el}$ represent the elemental expansion functions for the high-order spectral/*hp* DG method available in *Nektar++*.[16,17] Both the solution and test functions are discontinuous at the interface between elements.

Following the standard Galerkin procedure, a weak form of the mixed formulation (6)–(7) is obtained as follows. The discrete version of Equation (6) reads

$$\sum_{e=1}^{N_{el}} \int_{\Omega_e} w_i^e (\mathbf{g}_h^e - \nabla \mathbf{u}_h^e) d\Omega^e = \mathbf{0}; \quad i = 1, \ldots, N_{el}. \tag{9}$$

Using an approximation of the form (8) for both $\mathbf{u}_h^e$ and $\mathbf{g}_h^e$, and applying Gauss' theorem, this equation becomes

$$\sum_{e=1}^{N_{el}} \int_{\Omega^e} \left(w_i^e \sum_{j=1}^{N_{el}} \mathbf{g}_j^e w_j^e + \nabla w_i^e \sum_{j=1}^{N_{el}} \mathbf{u}_j^e w_j^e \right) d\Omega^e - \sum_{e=1}^{N_{el}} \int_{\Gamma^e} w_i^e \left(\sum_{j=1}^{N_{el}} \mathbf{u}_j^e w_j^e \right) \boldsymbol{n}\ d\Gamma^e = \mathbf{0}; \quad i = 1, \ldots, N_{el}, \tag{10}$$

where Γ^e denotes the boundary faces of the element Ω^e. The solution of this equation give us the discrete values of the first-order derivatives $\mathbf{g}_h^e$. To evaluate the integral expressions, we use an auxiliary mapping $\varphi_M : \boldsymbol{x} \mapsto \boldsymbol{\xi}$ to transform the local element coordinates $\boldsymbol{x}$=(x_1, x_2) to reference element coordinates $\boldsymbol{\xi} = (\xi_1, \xi_2)$ such that $-1 \leq \xi_1, \xi_2 \leq 1$ and all required operations take place in the reference element Ω_{st}, see Figure 1.

The weak form of Equation (7) is obtained in a similar fashion to give

$$\sum_{e=1}^{N_{el}} \int_{\Omega^e} w_i^e \sum_{j=1}^{N_{el}} \frac{d\mathbf{u}_j^e}{dt} w_j^e\ d\Omega - \sum_{e=1}^{N_{el}} \int_{\Omega^e} \nabla w_i^e \sum_{j=1}^{N_{el}} \mathbf{F}_j^e w_j^e\ d\Omega + \sum_{e=1}^{N_{el}} \int_{\Gamma_e} w_i^e \sum_{j=1}^{N_{el}} (\mathbf{F}_j^e \cdot \boldsymbol{n}) w_j^e\ d\Gamma = 0; \quad i = 1, \ldots, N_{el}. \tag{11}$$

The solution is discontinuous at the interface between the elements and the integrand in the boundary integral of Equation (11) is substituted by a numerical flux function. The convective normal flux at an interface is approximated by a numerical flux calculated via a Riemann solver

$$[(\mathbf{F}_c)_i^e \cdot \boldsymbol{n}]_{\Gamma_e} \approx \mathcal{H}^c(\mathbf{u}_e, \mathbf{u}_{e^+}; \boldsymbol{n}), \tag{12}$$

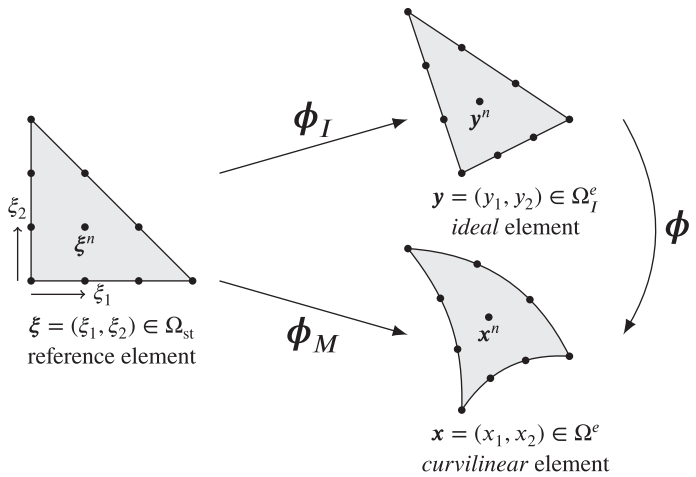


FIGURE 1 Existing isoparametric mappings between the reference, the *ideal* and the *curvilinear* elements. The *ideal* and *curvilinear* elements become, respectively, the *target* and *adapted* elements in the framework of *r*-adaptation

where $\mathbf{u}_{e^+}$ and $\mathbf{u}_e$ are the values of the conservative variables on the external and internal sides of the interface with respect to the eth element. This mechanism allows information to pass from one element to the other. The evaluation of the diffusive normal flux at the interface follows the local discontinuous Galerkin (LDG) formulation,[18] where it is approximated by

$$[(\mathbf{F}_d)_i^e]_{\Gamma_e} = \{\{\mathbf{F}_d\}\} + \mathbf{C}_{12}[[\mathbf{F}_d]] + C_{11}[[u]], \tag{13}$$

and similarly

$$[(\mathbf{u})_i^e]_{\Gamma_e} = \{\{\mathbf{u}\}\} - \mathbf{C}_{12}[[u]], \tag{14}$$

where $\mathbf{C}_{12} = \frac{1}{2}\mathbf{n}$, and C_{11} is an order 1 constant. The average and jump operators are defined as

$$\{\{u\}\} = \frac{1}{2}(u^+ + u^-), \quad [[u]] = (u^+\boldsymbol{n}^+ + u^-\boldsymbol{n}^-), \tag{15}$$

$$\{\{\mathbf{u}\}\} = \frac{1}{2}(\mathbf{u}^+ + \mathbf{u}^-), \quad [[\mathbf{u}]] = (\mathbf{u}^+ \cdot \boldsymbol{n}^+ + \mathbf{u}^- \cdot \boldsymbol{n}^-). \tag{16}$$

3.1 | Shock capturing via a discontinuity sensor

Our DG discretization of the Euler equations requires the addition of the diffusion flux, $\mathbf{F}_d$ to stabilize the solutions in the presence of shock waves. The term μ_a in Equation (2) is an artificial viscosity coefficient that allows dissipation to be selectively applied to shocks. For consistency $\mu_a \sim h/p$, we use[19]

$$\mu_a \sim \frac{h}{p}\lambda_{\max}, \tag{17}$$

where $\lambda_{\max} = |u| + c$ is the local maximum wave speed of the system. The characteristic element length h is chosen as the minimum edge length of an element. Finally, for the artificial viscosity to vanish outside shocks it needs to be proportional to a shock sensor, S, such as

$$\mu_a = \mu_0 \frac{h}{p}\lambda_{\max} S, \tag{18}$$

where $\mu_0 = O(1)$ is a constant. To build the shock sensor, we adopt a modal resolution-based indicator[8] which is elementwise constant and defined via an intermediary term

$$s_e = \log_{10}\left(\frac{\langle q - \tilde{q}, q - \tilde{q}\rangle}{\langle q, q\rangle}\right), \tag{19}$$

where $\langle\cdot\,,\cdot\rangle$ represents a L^2 inner product, *and* q and $\tilde{q}$ are the full and truncated expansions of a state variable given by

$$q(x) = \sum_{i=1}^{N(P)} \hat{q}_i\varphi_i, \quad \tilde{q}(x) = \sum_{i=1}^{N(P-1)} \hat{q}_i\varphi_i, \tag{20}$$

where φ_i are the basis functions, $\hat{q}_i$ the associated coefficients, and $N(P)$ the size of the expansion of order P. In our case the test variable q is chosen to be the density ρ. To spatially smooth out the variation of the values of the sensor, the constant elementwise sensor is computed as

$$S = \begin{cases} 0, & s_e \le s_0 - \kappa, \\ \frac{1}{2}\left(1 + \sin\frac{\pi(s_e - s_0)}{2\kappa}\right), & |s_e - s_0| \le \kappa, \\ 1, & s_e \ge s_0 + \kappa, \end{cases} \tag{21}$$

with $s_0 \sim \log_{10}(p^4)$ from an analogy to Fourier coefficients decaying as $1/p^2$, and κ needs to be sufficiently large to obtain a smooth shock profile. We select

$$s_0 = -s_\kappa - 4.25 \log_{10} p, \tag{22}$$

where s_κ and κ can be adjusted for a specific problem. We will describe how to choose these parameters in Section 6.

4 | *r*-ADAPTATION

In *r*-adaptation we are aiming to increase resolution by locally reducing the mesh size, h, while keeping the number of DOF in the mesh constant. This effectively requires us to cluster mesh nodes in the vicinity of those regions where additional resolution is required, for example, shocks. We propose to accomplish this by suitably adapting a variational framework for the optimization of high-order meshes.[20]

4.1 | Variational mesh optimization

The objective of this variational framework[20] is to improve the quality of high-order *curvilinear* elements by means of a node-based optimization approach using a formulation based on the energy of deformation. An important aspect of such energy-based formulation is that a suitable choice of the energy functional, namely one that is polyconvex, would guarantee the existence of a minimum and therefore of a solution to the minimization problem.

The deformation of the mesh and the displacements of its nodes are represented via *isoparametric* mappings as follows. Figure 1 shows that a mapping $\boldsymbol{\varphi}_M$ exists from a reference element $\boldsymbol{\Omega}_{st}$ to a *curvilinear* high-order element $\boldsymbol{\Omega}^e$. We can further decompose the mapping $\boldsymbol{\varphi}_M$ into two distinct mappings: a mapping $\boldsymbol{\varphi}_I$ from reference to *ideal* elements and a mapping $\boldsymbol{\varphi}$ from the *ideal* to the *curvilinear* elements. We define the *ideal* element as the high-order linear element, which after minimization will be the element that the optimizer seeks to achieve.

From this *ideal* element, we calculate the deformation energy. The mesh is deformed to obtain a new set of nodal coordinates that minimize an energy functional $\mathcal{E}(\nabla\boldsymbol{\varphi})$:

$$\text{find} \min_{\boldsymbol{\varphi}} \mathcal{E}(\nabla\boldsymbol{\varphi}) = \sum_e \int_{\boldsymbol{\Omega}^e} W(\nabla\boldsymbol{\varphi})\, d\boldsymbol{y}, \tag{23}$$

where $W(\nabla\boldsymbol{\varphi})$ is a formulation of the deformation energy. Several formulations were tested and it was found that the best results were obtained when using a hyperelastic model.[20] For this model, the strain energy takes the form

$$W = \frac{\mu}{2}(I_1^{\boldsymbol{C}} - 3) - \mu \ln J + \frac{\lambda}{2}(\ln J)^2, \tag{24}$$

where λ and μ are material constants, $\boldsymbol{C}$ is the right Cauchy–Green tensor, $I_1^{\boldsymbol{C}}$ is its trace, and J is the determinant of the Jacobian matrix $\boldsymbol{J} = \nabla\boldsymbol{\varphi}$. We use this formulation in the work that follows.

4.2 | Improving the shock resolution via element scaling

To achieve *r*-adaptation, we change this *ideal* element and make it an arbitrary *target* element.[5] The optimizer now aims to *adapt* an element $\boldsymbol{\Omega}^e$ toward a size and a shape similar to the *target* element $\boldsymbol{\Omega}_T^e$ (see Figure 2). Previous work[5] has shown that performing *r*-adaptation on the high-order mesh yielded spurious deformations. For this reason, *r*-adaptation, in this work, is performed on the linear mesh before

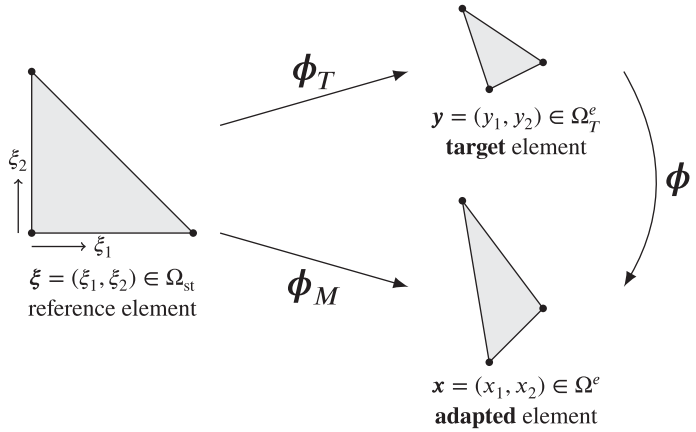


FIGURE 2 Existing mappings between the reference, the *target* and the *adapted* elements for *r*-adaptation

it is enriched to high order. This allows us to obtain high-order meshes of good quality, even after *r*-adaptation.

Although in principle the *target* element $\boldsymbol{\Omega}_T^e$ can take any shape and size, we have adopted a practical approach in this work that aims at avoiding too large deformations. The rationale for this is that the definition of a *target* element $\boldsymbol{\Omega}_T^e$ that is very different from the *ideal* element $\boldsymbol{\Omega}_I^e$—that is, the initial linear element before *r*-adaptation—introduces extra energy in the system that the optimizer has to minimize and thus slows down the process. For this reason, we define a *target* element $\boldsymbol{\Omega}_T^e$ with respect to the *ideal* element $\boldsymbol{\Omega}_I^e$. The *ideal* element $\boldsymbol{\Omega}_I^e$ can be manipulated anisotropically[5] by applying a metric tensor $\boldsymbol{M}$ to the Jacobian of the mapping, $\boldsymbol{J} = \nabla\boldsymbol{\varphi}$. We transform the *ideal* element $\boldsymbol{\Omega}_I^e$ into the *target* element $\boldsymbol{\Omega}_T^e$ through

$$\boldsymbol{J}_T = \boldsymbol{M}\boldsymbol{J}_I. \tag{25}$$

We do not consider directionality in this work and only *shrink* elements where additional resolution is required, that is, in the shock regions. In this case, the Jacobian is simply scaled by a linear shrinking factor r_{scale}. The metric tensor M is simplified to $r_{\text{scale}}\boldsymbol{I}$ to obtain

$$\boldsymbol{J}_T = (r_{\text{scale}}\boldsymbol{I})\boldsymbol{J}_I. \tag{26}$$

This framework was implemented in *NekMesh*, an open-source software solution for the generation of geometry-accurate high-order meshes, part of the *Nektar++* platform.[16,17]

5 | *p*-ADAPTATION

To enhance resolution in regions of smooth flow through local *p*-adaptation, we use the following fairly straightforward procedure.[4,14] We increase the local resolution by increasing the polynomial order within the elements where the local error is estimated to be high and we decrease the local resolution or, equivalently, the elemental polynomial order within those elements where the local error is estimated to be low.

We summarize this procedure in Algorithm 1 where e denotes an individual element, s_e and p_e are its associated error indicator and polynomial order, ε_u and ε_l are the upper and lower error thresholds, and $p_{\max}$ and $p_{\min}$ are the maximum and minimum polynomial orders allowed.

At every iteration (see outer loop), the polynomial order of individual elements is increased or decreased by 1 single order at most. It may also remain the same if neither the upper nor the lower threshold is reached. Following this change in element polynomial orders, we project the steady-state solution of the previous iteration onto the new polynomial space and the process continues. We use the formulation of sensor s_e in Equation (19) to drive the local change in polynomial order.

Algorithm 1. The *p*-adaptive procedure

```
repeat
    calculate the steady-state solution
    for all e do
        calculate s_e
        if s_e > ε_u and p_e < p_max then
            increment p_e
        else if s_e < ε_l and p_e < p_min then
            decrement p_e
        else
            maintain p_e
        end if
    end for
until no p_e is modified
```

6 | WORKFLOW FOR *RP*-ADAPTATION

Finally we attempt to combine the best properties of the two previous strategies in the same simulation to maximize their effect in increasing the resolution of compressible flow simulations. More specifically, *r*-adaptation will be responsible for the resolution of shocks, whereas *p*-adaptation will resolve smooth flow regions. In the proposed *rp*-adaptation workflow these adaptative techniques will be alternatively applied in a sequence of steps that is described in the following.

As noted in Section 1, the sensor of Equation (19) is used concurrently to add artificial viscosity, to move mesh DOF toward shocks and to drive local *p*-adaptation. We emphasize that although the original intention of this sensor in the work of Persson and Peraire[8] was to identify regions of discontinuity, in this work we will also use the sensor to drive a *p*-adaptation process. This is a valid strategy since the sensor is ultimately a resolution indicator that is based on the decay of modal energy, which therefore makes it applicable for usage in general regimes, and not only those that contain shocks. This is directly validated by the work of Naddei et al.[11] In addition, indicators based on this concept have been successfully leveraged in both spectral element solvers of smooth flows, where the efficacy is demonstrated from the context of incompressible *h*-adaptivity,[12] as well as a broader numerical context of, for example, high-order finite difference simulations of smooth compressible flows.[13]

We first generate an initial high-order mesh for the domain. We anticipate the requirements of *r*-adaptation and the need for DOF to be moved around when deforming the mesh. For this reason, we generate a relatively coarse mesh, but with enough resolution to allow for the movement of mesh nodes. We then proceed to run the solver on this initial mesh and obtain a flow solution which represents our base solution. During this step, we have to determine appropriate parameter values for the artificial viscosity.

The artificial viscosity term (2) depends on three parameters: μ_0, s_κ, and κ. As it is common practice in codes based on artificial viscosity, we start with *Nektar++* default values, and then tune the parameters for our specific problem. The default values are determined via one-dimensional analysis of the scaling and smoothing of the solution by its modal decay, as described for instance by Klöckner et al.[21] The level of artificial viscosity (μ_0) is chosen empirically to be sufficiently large so as to obtain a sharp but smooth shock profile and it is tuned so that the shock is stable but not overly dissipative. In the sensor Equation (19), the parameter κ is the width of the activation window of shock sensor values and the parameter s_κ, together with κ, sets the threshold above which the shock is detected. The artificial viscosity parameters (s_κ and κ) are adjusted to ensure that artificial viscosity is only triggered in the direct vicinity of shock waves, by excluding values of large gradients elsewhere, for example, near stagnation points or trailing wakes. The final values of the tuned parameters are given in Section 7 for the two test cases considered in this work.

From this base solution, we apply *r*-adaptation to the mesh. We first extract the list of elements where artificial viscosity was added during the initial simulation. If the run was set up properly, these elements only represent the regions where a shock is present. From these elements, we extract their barycentres and assign an isotropic shrinking

factor r_{scale} to them (see Section 4.2). This shrinking factor r_{scale} is currently chosen empirically in such a way that as many nodes as possible are pulled inside the shock area, without compromising the quality of the mesh. From experience, values as low as $r_{\text{scale}} = 0.1$ can be used in open geometries while more moderates values must be used in closed geometries. Both situations are illustrated below. For all the other elements, we also extract the barycentres and assign a factor $r_{\text{scale}} = 1$. In practical terms, we force elements in the shock regions to shrink and pull mesh nodes from other parts of the mesh. This field of r_{scale} factors is then supplied to the variational *r*-adaptation code which is then run on the linear mesh. The variational framework optimizes the mesh so that each element is as close as possible to its target size, effectively moving nodes from areas of $r_{\text{scale}} = 1$ to areas of $r_{\text{scale}} < 1$. We also note that *r*-adaptation is run on the linear mesh before making the adapted mesh high order again. This significantly speeds up the optimization procedure and improves the validity of the final mesh. We then run the solver on the adapted high-order mesh and obtain a new solution with enhanced shock resolution. This procedure can optionally be repeated based on the new solution.

From this solution on the adapted mesh, we can run *p*-adaptation as described in Section 5. At the end of each cycle, a sensor value is computed for each element and the local polynomial order of that element is decreased, kept the same or increased based on the value of the sensor. Throughout this work we use *Nektar++* default values for the adaptation parameters: $\varepsilon_u = -6$, $p_{\max} = 6$, $\varepsilon_l = -8$, and $p_{\min} = 2$. The simulation then proceeds onto a new cycle and the process is repeated until a steady solution is obtained and the local polynomial orders do not vary.

We also allow the user a choice to restrict the polynomial order of elements within the shock regions. These are the zones that have been previously identified in the *r*-adaptation procedure. The local polynomial order of the elements in these regions is then set to a user-defined value, which should be typically low ($p < 3$). The proposed *rp*-adaptation workflow is summarized in Algorithm 2.

Algorithm 2. The *rp*-adaptive workflow

```
generate an initial high-order mesh
calculate the steady-state solution
repeat
    extract shock areas based on sensor values
    apply r-adaptation in shock areas to linear mesh and re-project to high-order
    calculate the steady-state solution
until shocks are well captured
apply p-adaptation as described in Algorithm~1
calculate the final solution
```

7 | NUMERICAL EXAMPLES

In this section we use two different test cases to demonstrate the *rp*-adaptation workflow: a NACA 0012 aerofoil in transonic regime and a supersonic intake. Different difficulties arise for each of these test cases as we will discuss below. Most importantly, we take slightly different approaches when it comes to *r*-adaptation and *p*-adaptation.

The transonic flow in the first test develops two shocks: a strong shock on the upper surface of the aerofoil and a weak shock on the lower surface. The disparity of strengths between the two shocks permits us to verify the shock-capturing ability of the artificial viscosity. From the point of view of the adaptative procedures, the purpose of the test case is twofold. In the first instance, we chose a fine mesh for the flow simulation and run a single round of *r*-adaptation to show that the variational framework can easily pull mesh nodes from the smooth regions toward the shock and that, in our implementation, the movement of the nodes is compatible with the computer-aided design (CAD) definition of the boundaries of the computational domain. Furthermore, our choice of a fine mesh allows us to use this flow simulation as an accuracy benchmark for subsequent simulations. In the second instance, we use this test case to analyze the effectiveness of our order restriction strategy in *p*-adaptation. We study three different approaches to order restriction. In the first approach, no restriction is applied and the local polynomial order of elements in shock areas is left free to increase. In the second one, we keep the local polynomial order of these elements constant, that is, the order of the initial simulation. In the third and final one, we immediately decrease the local polynomial order of these elements to the minimum allowed in the run.

In the second test, the supersonic inflow ($M_\infty = 3$) at the entrance of an intake generates a complex diamond-like pattern of oblique shocks inside the intake due to reflections of the shocks at its internal walls and their interactions. In this example, the number of nodes available for deforming the mesh is limited, which places additional stress on the variational optimization process. We therefore decide to take a two-step approach to *r*-adaptation where each step is run with a milder shrinking factor r_{scale} in order to retain good mesh quality.

7.1 | NACA 0012 aerofoil

We first demonstrate the new technology on a canonical aeronautical test case: a transonic ($M_\infty = 0.8$) inviscid flow past a NACA 0012 aerofoil at 1.25° angle of incidence. This configuration produces two shocks:[22] a strong shock on the suction side and a weak shock on the pressure side at approximately 60% and 35% of the chord, respectively. This provides a relatively easy test case to showcase the technology where the shocks are quasi-vertical. The main difficulty lies in the relative weakness of the shock on the pressure side and in capturing it appropriately.

The domain used has external boundaries at a distance of 40*c* from the aerofoil, where *c* denotes the aerofoil's chord length. We discretize the domain uniformly along the chord with an element of size approximately 0.5*c* on the aerofoil boundary and create a smooth progression toward an element of size approximately 10*c* on the outer boundary. The automatic sizing of elements in the field is determined through an octree system.[23] The mesh is curvilinear of order $p = 4$ and it is optimized using the variational framework described in Section 4.1. Figure 3(A) (left) shows what the mesh looks like in the near field. The starting mesh is relatively coarse but it is run through the solver at uniform $p = 4$ order.

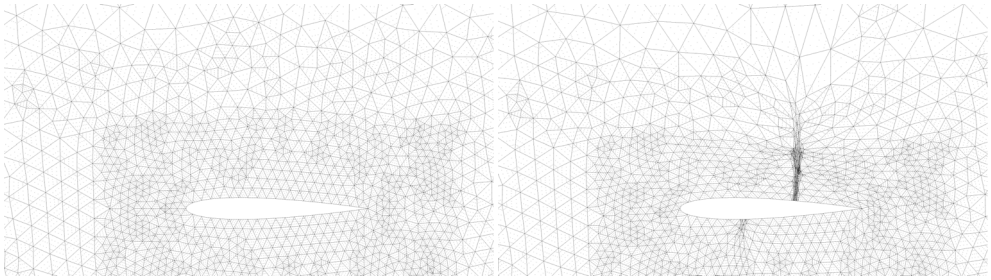
It is important to note the importance of having sufficient resolution (either through *h* or *p*) in the initial mesh in order to distinguish shocks, that is, actual discontinuities, from smooth high-gradient regions when looking at high discontinuity sensor values. The necessity of having a sufficient number of DOF in the mesh arises from the nature of the numerical representation of shocks. More specifically, shocks are captured over a fixed number of elements in the mesh that depends on the resolution capabilities of the particular CFD algorithm employed. The requirement of enough resolution in the initial mesh is general and applies to any type of adaptivity strategy used. Although one may sensibly consider including other forms of adaptivity (e.g. *h*- or *p*-adaptation) before improving the resolution of shocks using *r*-adaptation only, these considerations are beyond the scope of this article, which looks at the feasibility of the novel proposed adaptation strategy.

We first run the solver on the initial mesh to obtain a base solution. We impose slip wall BC on the surface of the profile and far-field BC at the external boundaries of the domain. We use the Harten–Lax–van Leer-contact (HLLC) Riemann solver.[24] For the artificial viscosity, we tuned the solver parameters to $s_\kappa = -1.2$, $\kappa = 0.7$, and $\mu_0 = 1.0$. Figure 3(C) (left) shows that large values of the sensor are obtained in both shock areas but also near the leading and trailing edges. However, Figure 3(D) (left) shows that artificial viscosity only triggers in the vicinity of the two shocks, proving adequate tuning of the artificial viscosity parameters. The flow solution in Figure 3(B) (left) displays very thick shocks as expected on this relatively coarse mesh. We can also observe oscillations in the field past the strong shock caused by the underresolution of the shock and the generation of entropy.

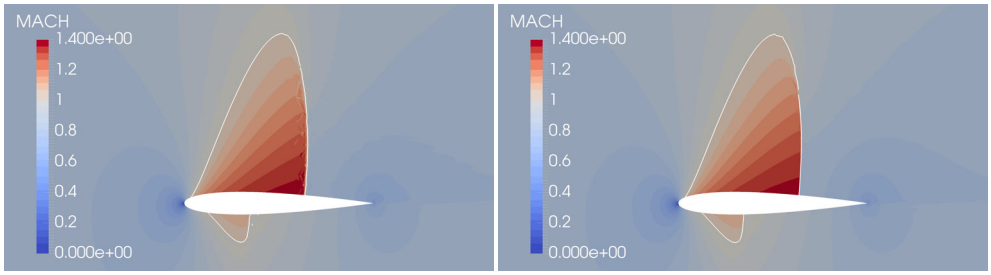
7.1.1 | *r*-adaptation

From the base solution, we follow the workflow explained in Section 6. We first extract the shock regions: these correspond to the elements of non-zero artificial viscosity in Figure 3(D) (left). To these regions, we assign a shrinking factor $r_{\text{scale}} = 0.1$ and run the *r*-adaptation procedure. We obtain a new mesh which, for quality considerations, we reoptimize before simulation. The new mesh shown in Figure 3(A) (right) shows refinement in the shock areas and consequently a slight coarsening outside of those zones. Shrinking is also observed, to a smaller extent, in the vertical direction due to the isotropy of the *r*-adaptation approach. However, the resulting mesh is clearly anisotropic and aligned to the presence of the shock.

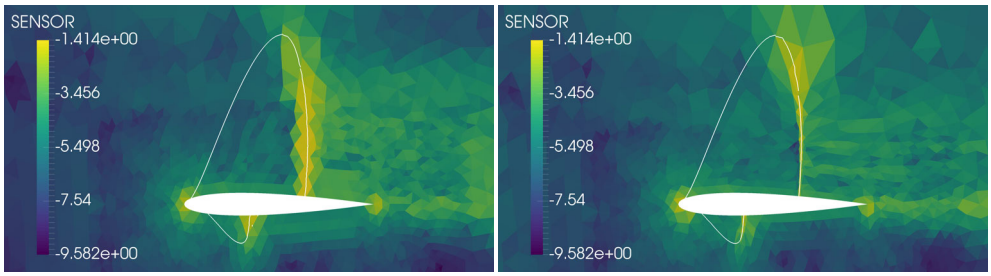
We now interpolate the old solution onto the adapted mesh and run the simulation again. In order to avoid any instability of the solver due to the interpolation of the underresolved shock onto the new mesh, we first run the solver over a few hundred time steps with a decreased step size. We then run the simulation, using the exact same artificial viscosity parameters, until a steady state is achieved. The flow solution in Figure 3(B) (right) shows better resolution of both shocks as seen by the sharpness of the shocks. We also observe reduced oscillations in the wake of the strong shock.



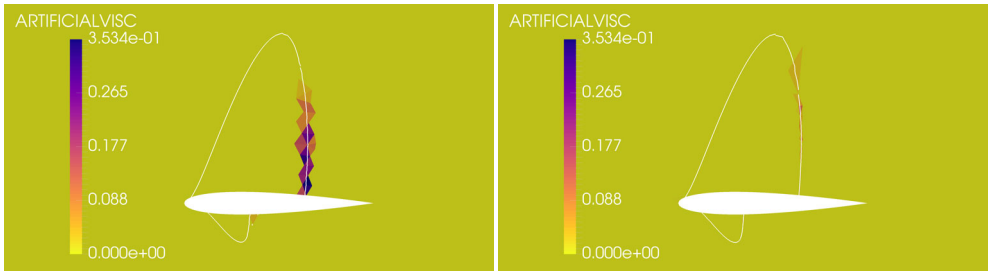
(A) Mesh.



(B) Mach number field.



(C) Sensor field.



(D) Artificial viscosity field.

FIGURE 3 Comparison of the mesh and fields for the transonic NACA 0012 aerofoil test case before (left) and after (right) *r*-adaptation. A white line denotes the *sonic* line $M = 1$ in Figure 3(B–D) [Color figure can be viewed at wileyonlinelibrary.com]

Figure 3(C,D) (right) finally shows that discontinuity, as per the sensor, is now observed in a narrower area and that the artificial viscosity reaches lower values.

The improvement of the resolution of the shock can be better seen by visualization of quantities along the wing surface. Figure 4 shows the Mach number on the surface of the profile. Because elements are so large in the initial mesh, the solution shows strong oscillatory behavior, known as the Gibbs phenomenon described in Section 1, inherent to the low-dissipation discretization used in this work. The artificial viscosity term (2) introduced to stabilize the solution reduces this phenomenon but does not totally eliminate it. At this point, we do not seek to further increase the artificial viscosity, which would cause more dissipation. Instead, our goal is to simply stabilize the solution.

After *r*-adaptation, elements in the region of the shock are much smaller and, although oscillations are still present, they have both a smaller amplitude and a narrower range, thanks to increased resolution in the region. This confirms the qualitative observation of the increased sharpness of the shock seen in Figure 3(B). Because there is less mesh movement at the weak shock, the reduction in the Gibbs phenomenon is also smaller.

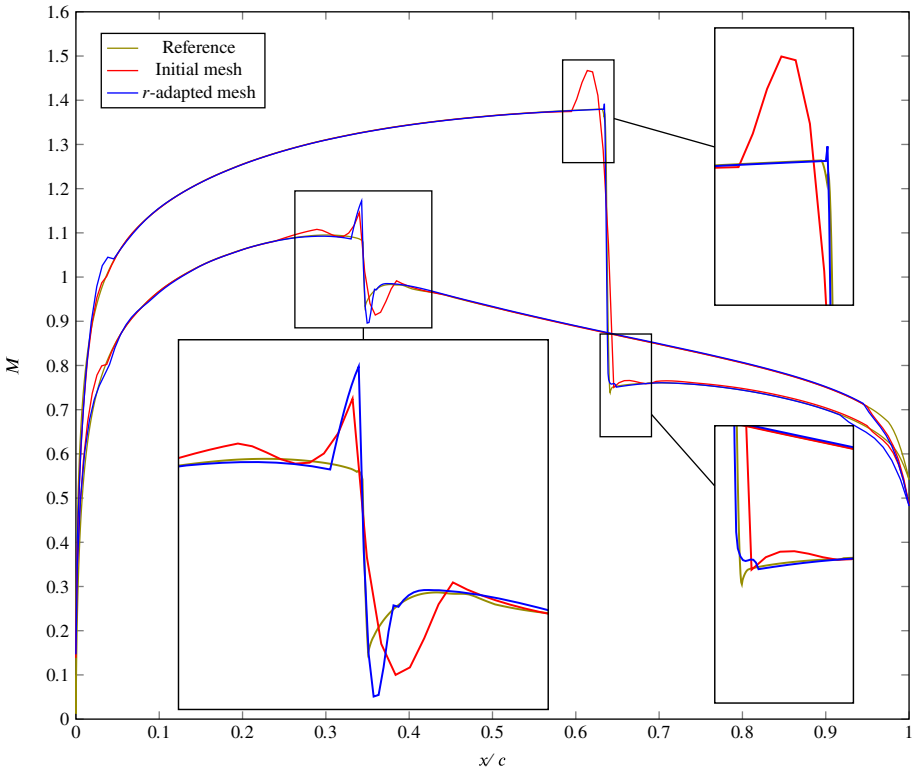


FIGURE 4 Transonic NACA 0012 aerofoil: Plot of the Mach number M before and after r-adaptation [Color figure can be viewed at wileyonlinelibrary.com]

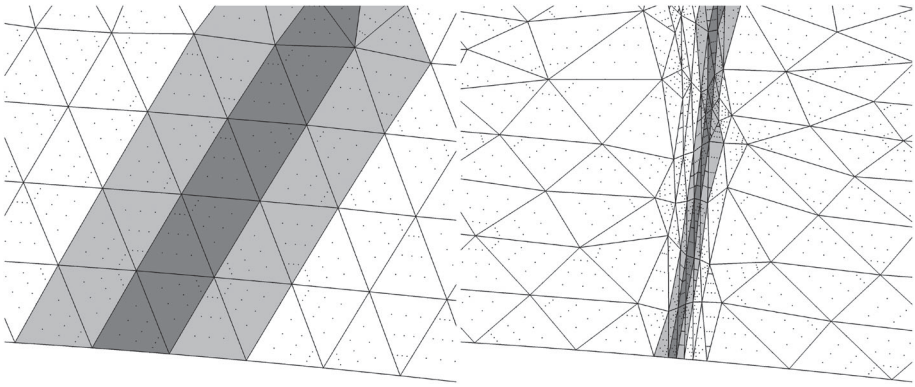
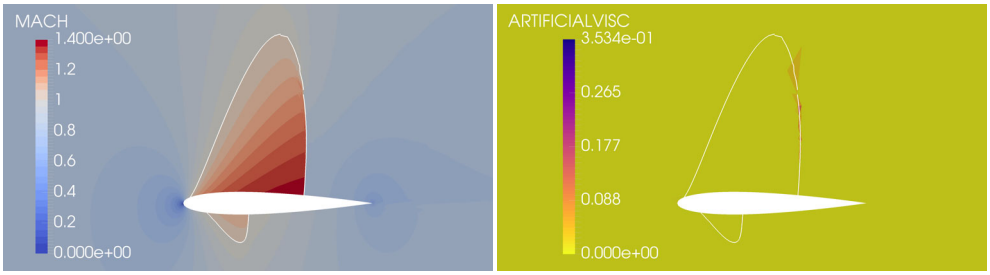


FIGURE 5 Magnified view of boundary elements with *CAD sliding* enabled nodes for the NACA 0012 aerofoil test case: original (left) and adapted (right) meshes. Three rows of elements have been colored to highlight the large movement of nodes

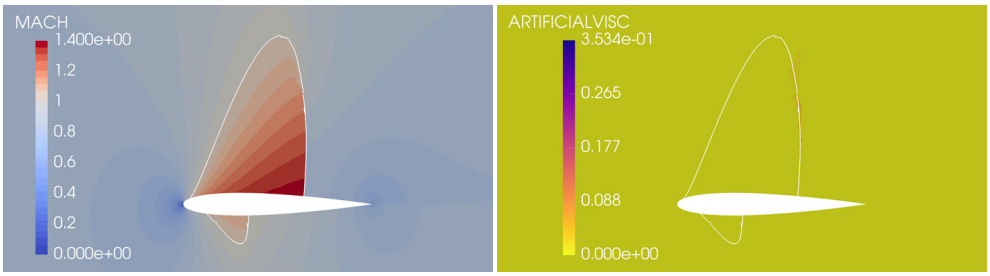
We also draw the reader's attention to the importance of the presence of CAD data for the refinement process, as in order to retain accurate boundary representation, it is important that the r-adaptation code has access to a CAD system. In this instance, *NekMesh*[23] and the variational optimizer have been implemented to use the OpenCASCADE[25] framework as its CAD engine. This allows *NekMesh* to query the geometry and ensure that all nodes remain on the boundaries at all time. The capability to slide nodes along the surface (*CAD sliding*) is shown in Figure 5 where nodes remain on the aerofoil surface throughout the r-adaptation process. Figure 5 also shows that the optimizer is able to move nodes across large distances, as seen through the row of colored elements before (left) and after (right) r-adaptation.

7.1.2 | *p*-adaptation

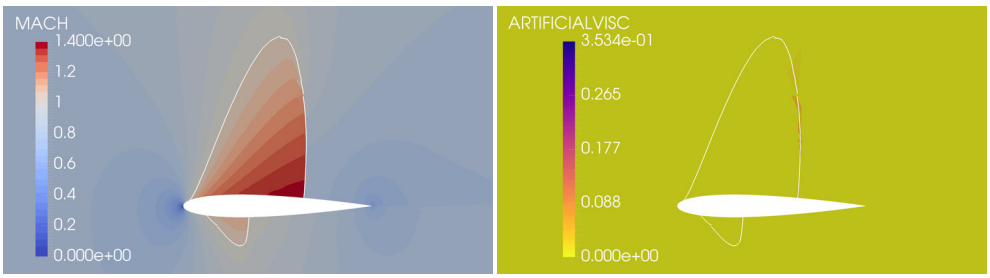
After better resolving the shocks, we can now apply local p-adaptation for the smooth field. For this test case, we compare three scenarios. We apply local p-adaptation without any restriction in the first scenario (see Figure 7(A)) while, in the other two scenarios, we restrict the local polynomial order inside the shock areas. In the second scenario, we preserve the local polynomial order of the uniform $p = 4$ order simulation of Section 7.1.1 (see Figure 7(B)). In the



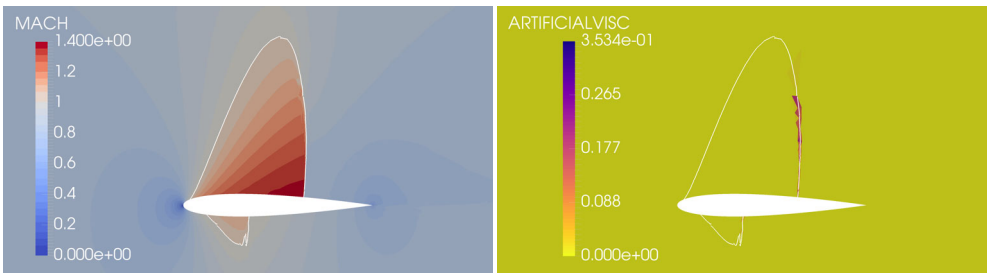
(A) Uniform p.



(B) Full p-adaptation.



(C) p-adaptation with original order restriction.



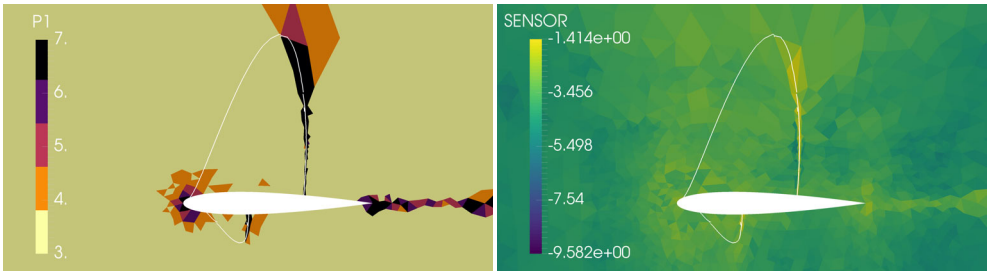
(D) p-adaptation with lowest order restriction.

FIGURE 6 Comparison of the Mach number (left) and artificial viscosity (right) fields for the uniform p simulation and the three test scenarios of the transonic NACA 0012 test case. A white line denotes the sonic line ($M = 1$) in all figures [Color figure can be viewed at wileyonlinelibrary.com]

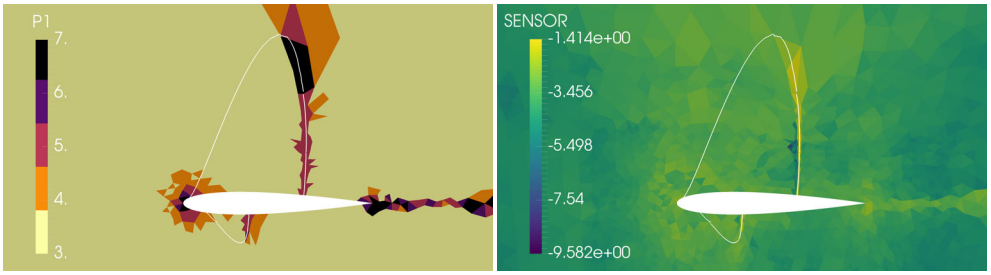
third and last scenario, we decrease the local polynomial order inside the shock areas to the lowest user-allowed order (see Figure 7(C)).

For these tests, we start from the field obtained at $p = 4$ in Section 7.1.1 and use values of $p_{\min} = 2$ and $p_{\max} = 6$. The sensor is based on the density field ρ and the solver default values of lower and upper sensor tolerances are respectively 10^{-8} and 10^{-6}. Figure 7 shows the results. The figures on the left show a final map of the local number of modes ($=p+1$) after a steady solution is reached and, by extension, when the local polynomial order remains constant throughout p-adaptation steps. The number of DOF for each simulation is shown in Table 1. All scenarios produce fewer DOF than the simulation at uniform p, thanks to local p-coarsening in low-error regions. By design, the unrestricted p-adaptation scenario increases the local polynomial order of elements in the shock region to the maximum user-allowed value. This leads to a higher global number of DOF than the other two scenarios. Then follows the second scenario while the last scenario has the smallest number of DOF. Each of these DOF counts also translates into similar increases or decreases in computing times.

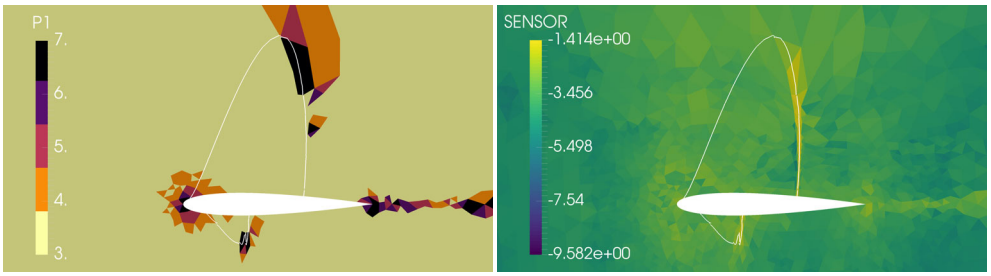
We compare these solutions to a reference solution computed on a very fine mesh. To evaluate the performance of each mesh and p-adaptation scenario, we look at the Mach number distribution on the surface of the aerofoil. We use the



(A) Full p-adaptation.



(B) p-adaptation with original order restriction.



(C) p-adaptation with lowest order restriction.

FIGURE 7 Comparison of the local number of modes ($=p+1$; left) and postadaptation sensor fields (right) for the three test scenarios of the NACA 0012 aerofoil test case. A white line denotes the *sonic* line ($M = 1$) in all figures [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Number of DOF, error, and CPU time per time step at convergence for the transonic NACA 0012 aerofoil test case

		$\|\|e\|\|^2_{L^2(S)}(10^{-4})$			
Simulation	**Number of DOF**	**Pressure**	**Suction**	**Total**	**CPU time (ms)**
Initial mesh	65 550	0.547	5.28	5.83	65
r-Adapted mesh	65 550	0.751	1.36	2.11	118
Scenario #1	29 201	0.875	2.54	3.41	48
Scenario #2	29 117	0.919	1.82	2.74	56
Scenario #3	27 736	1.045	6.61	7.65	45

L^2-norm of the error, defined as

$$\|e\|^2_{L^2(S)} = \int_S (M - M_{ref})^2 \, dS,$$

where M is the Mach number of the test solution, M_{ref} is the Mach number of the reference solution, and S is the chord. Results are reported in Table 1. Note that central processing unit (CPU) times per time step are reported as run on a 16-core machine, once convergence is reached. We first note that r-adaptation alone provides an important boost in terms of accuracy. Scenarios #1 and #2 both suffer a loss of accuracy due to the coarsening of the solution in large parts of the domain. This slight increase in the error, however, allows us to reduce the number of DOF in half. Scenario #3, on the other hand, performs very poorly, with the error going even higher than on the original mesh. Decreasing the polynomial order inside the shock—a rather small region—allows us to save a few more DOF but at too great a cost.

Figure 6 shows a comparison of the Mach number (left) and artificial viscosity (right) fields for the uniform p simulation and the three test scenarios. We observe little difference between scenarios #1 and #2. Scenario #3 on the other hand exhibits underresolution of the shock, seen through its thicker profile. This is consistent with the local element size and the lack of DOF in the thickness of the shock at lower order. As a result, the last lower order scenario exhibits some oscillations in the wake, due to the generated entropy in the shock area. We can also observe that lower order scenarios induce more artificial viscosity. This phenomenon is consistent with the previous assessment of the lack of resolution of the shock. The discontinuity sensor detects a certain lack of resolution and therefore more artificial viscosity is added to the system.

Overall, all scenarios expectedly exhibit similar distributions of local polynomial order in the smooth field regions in Figure 7. When analyzing the distribution of local polynomial orders, we observe higher orders in the area above the strong shock and below the weak shock, in all scenarios. These areas were not detected as part of the shock in Section 7.1.1 because they were underresolved and therefore too short. Now that the shocks are better resolved, they reach further out and require additional resolution, in the form of higher polynomial order in this case. We also observe that parasite higher order zones are created in the lower order scenarios. This is especially obvious around the weak shock in the third scenario. As we noted above, this is due to the thicker shock profile and therefore the need to add resolution around the shock. Figure 7(C) (left) is consistent with this explanation as we observe a larger area of high sensor values, extending beyond the shock areas determined in Section 7.1.1.

7.2 | Supersonic intake

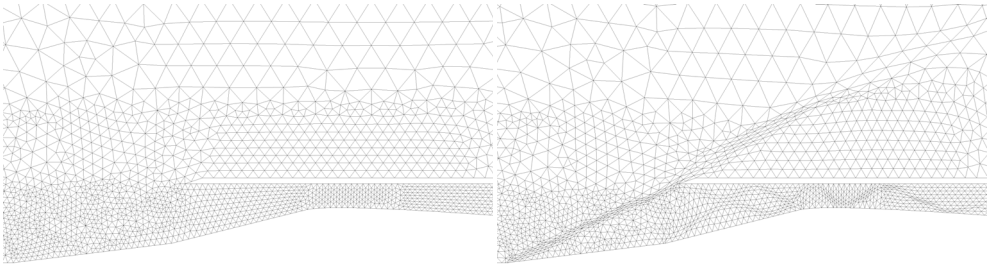
This section illustrates the new approach on a test case with a more complicated shock pattern. The test case is that of a supersonic intake at $M_\infty = 3.0$ first studied experimentally[26] and later numerically.[27] The intake consists of two straight ramps inclined with respect to the incoming free-stream flow at angles of 7° and 14°, respectively. The first ramp creates an oblique shock which impinges on the horizontal cowl and in turn leads to a complex pattern of reflecting oblique shocks throughout the diffuser of the intake. The difficulty here is the presence of multiple shocks with different orientations in the very narrow regions of the diffuser.

We discretize the domain uniformly in the streamwise direction. We set an element of size $0.01L$ (L being the length of the intake) inside the intake and let it coarsen outside the intake up to an element of size $0.05L$ in the far-field. The mesh is curvilinear of order $p = 4$ and it is optimized in the throat. Figure 8(A) (left) shows what the mesh looks like inside the intake and in its immediate surroundings.

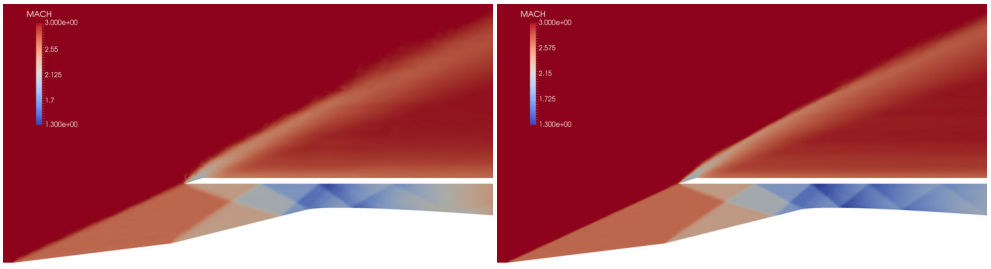
We run the solver at uniform order $p = 3$ on the initial mesh to obtain a base solution. We impose wall BC on the surfaces of the intake, at the intake outlet BC we set a low enough pressure until a fully supersonic field is obtained ($P_b = 0.9P_{\text{inf}}$) and far-field BC at the external boundaries of the domain. We use Roe's approximate Riemann solver.[24] For the artificial viscosity, we tuned the solver parameters to $s_\kappa = 0.0$, $\kappa = 0.0$ and $\mu_0 = 0.1$. Figure 8(C) (left) shows that large values of the sensor are obtained in all shocks and that moderate values are obtained everywhere after the first upstream shock. However, Figure (left) shows that the artificial viscosity is only triggered in the vicinity of the shocks, proving that the tuning of the artificial viscosity parameters is adequate. Just like for the NACA 0012 test case, the shocks exhibit a thick profile, as can be seen in Figure 8(B) (left), due to the relatively coarse local mesh as well as some oscillations near the leading edge of the cowl.

7.2.1 | *rr*-adaptation

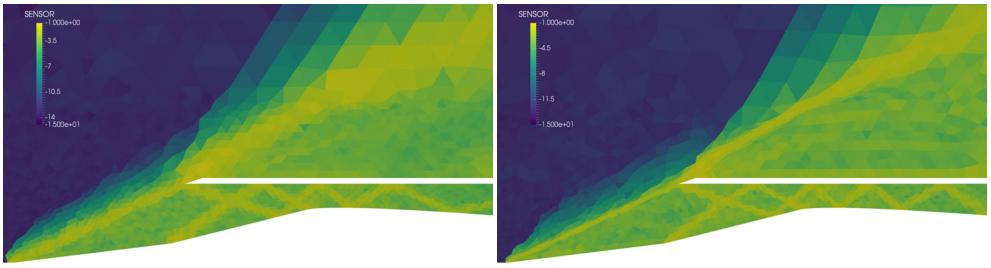
Once more we follow the workflow laid out in Section 6 except that we decide to run two rounds of r-adaptation. Each round uses a less aggressive shrinking factor $r_{\text{scale}} = 0.5$. Before each simulation, we again carry out mesh optimization to improve the quality of the high-order mesh. The new mesh after one round of r-adaptation is depicted in Figure 8(A) (right). We observe refinement in all areas of interest and note that refinement is stronger in the area of the first upstream shock. Indeed, elements in the first shock are able to pull DOF from the freestream areas, whereas elements inside the intake are interacting with each other. Refinement is nonetheless obtained in all shock areas and anisotropy naturally appears such that elements are shrunk mostly in the direction normal to the shock. The r-adaptation strategy works by pulling nodes together. Even though the optimizer is not aware of the shock structures, nodes are naturally moved normally rather than tangentially to the underlying shock, because the shrinking areas are long and narrow.



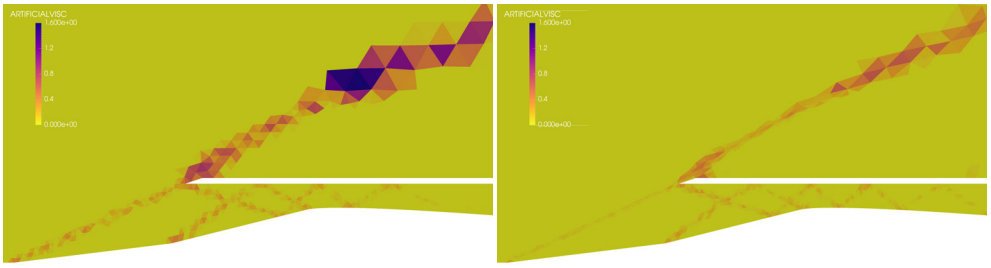
(A) Mesh.



(B) Mach number field.



(C) Sensor field.



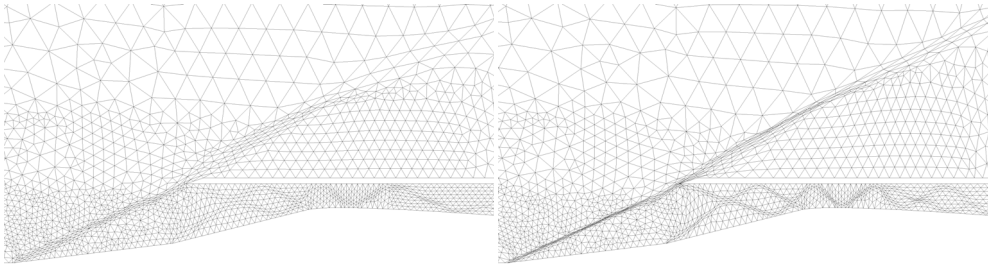
(D) Artificial viscosity field.

FIGURE 8 Comparison of the mesh and fields for the supersonic intake before (left) and after (right) the first round of *r*-adaptation [Color figure can be viewed at wileyonlinelibrary.com]

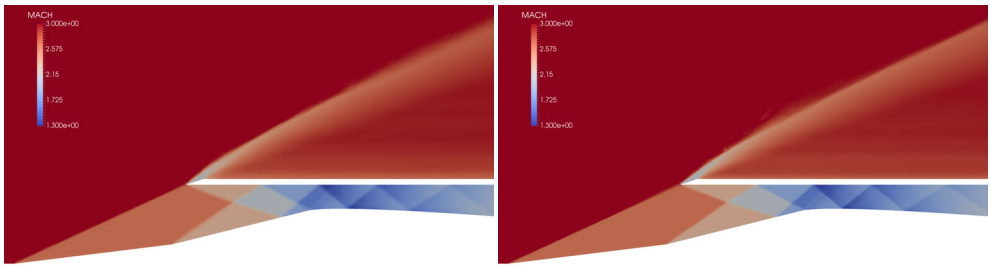
We now run the solver on the new adapted mesh using the same solver parameters. A stable flow solution is obtained and shown in Figure 8(B) (right). All shocks now appear sharper and the oscillations observed near the leading edge of the cowl have disappeared. Figure 8(C,D) (right) also shows that the discontinuity, as per the sensor, occurs in a narrower region.

We then apply a second round of *r*-adaptation in the exact similar fashion: we isolate shock areas and use them as input for the optimizer. Figure 9(A) (right) depicts the final adapted mesh which shows further refinement of the shock regions. We also notice that the oblique shocks inside the intake past the throat have moved upstream due to the refinement of the oblique shocks located upstream of the throat. While the *r*-adapted mesh could not capture these downstream shocks, the *rr*-adapted mesh can. By using a two-step approach, we are also able to pull more mesh nodes together than when using a one-step approach.

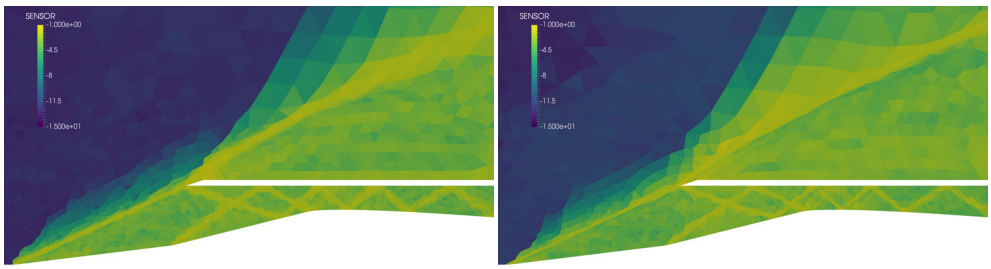
This becomes even more obvious when looking at the plot of the Mach number in Figure 10. The initial mesh largely overestimates the values of the Mach number past the throat ($x/L \approx 0.57$). This is mostly solved by *r*-adaptation although further improvement is obtained through *rr*-adaptation. This is due to the good resolution of upstream shocks through the clustering of DOF.



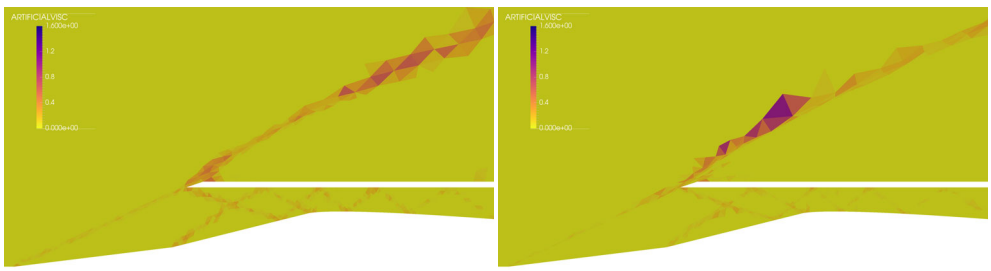
(A) Mesh.



(B) Mach number field.



(C) Sensor field.



(D) Artificial viscosity field.

FIGURE 9 Comparison of the mesh and fields for the supersonic intake before (left) and after (right) the second round of *r*-adaptation [Color figure can be viewed at wileyonlinelibrary.com]

7.2.2 | *p*-adaptation

We now apply *p*-adaptation to the *rr*-adapted mesh. For this test case, we employ unrestricted *p*-adaptation where the local polynomial order inside elements is left free to change, even in shock areas. We start from the field obtained at $p = 4$ in Section 7.2.1 and use values of $p_{\min} = 2$ and $p_{\max} = 6$. We again use a sensor based on the density field ρ and solver default values for the thresholds.

First, we observe that no steady state is achieved. Upon inspection, we notice that the system jumps back and forth between two states at each *p*-adaptation cycle. The two states correspond roughly to coarser and finer resolved fields. In the coarser resolved state, sensor values in shock areas are high. At the end of the *p*-adaptation cycle, these large sensor values trigger an increase in local polynomial order of a number of elements. Simulation goes on and the finer resolved state is obtained where sensor values are low. This in turn triggers a decrease in local polynomial order of the same elements, returning the system to the former coarser resolved field. This is shown in Figure 11 with the coarser resolved state on the left and the finer one on the right. We explain this behavior by a naive *p*-adaptation approach using simple sensor thresholds. The problem is highly nonlinear and nonlocal and error from refining/coarsening regions propagates along characteristics. The nonadjoint nature of the refinement strategy is bound to produce this sort of behavior.

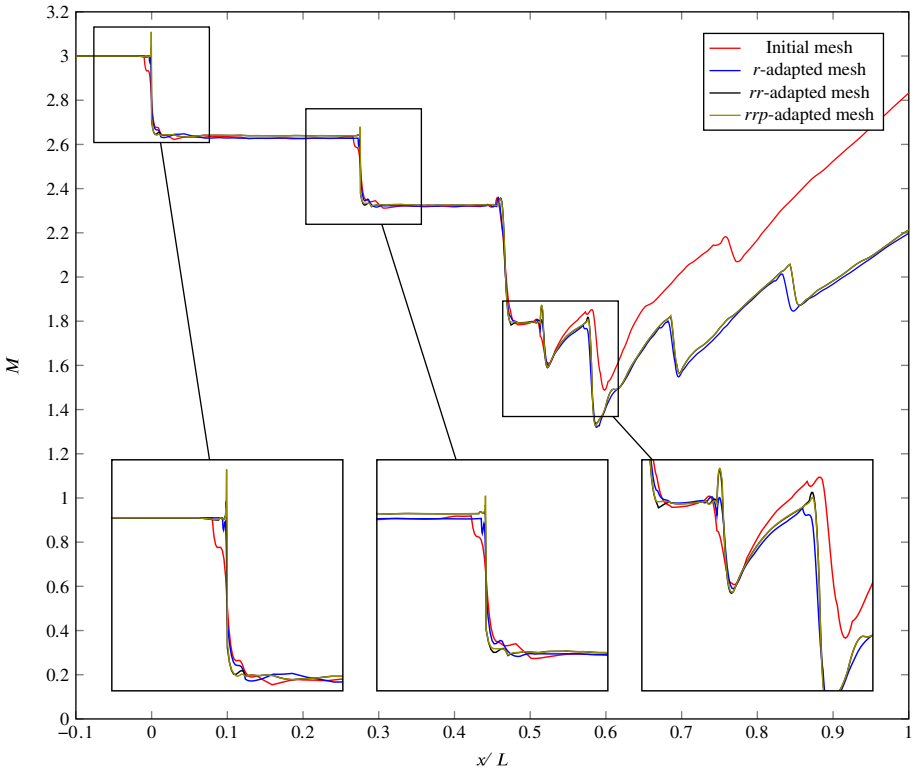


FIGURE 10 Supersonic intake: Plot of the Mach number M on the lower surface throughout the *rrp*-adaptation process [Color figure can be viewed at wileyonlinelibrary.com]

Simulation	State	Number of DOF	CPU time (ms)
Uniform p		40 210	27
Unrestricted p-adaptation	Finer	39 527	74
Unrestricted p-adaptation	Coarser	36 696	69

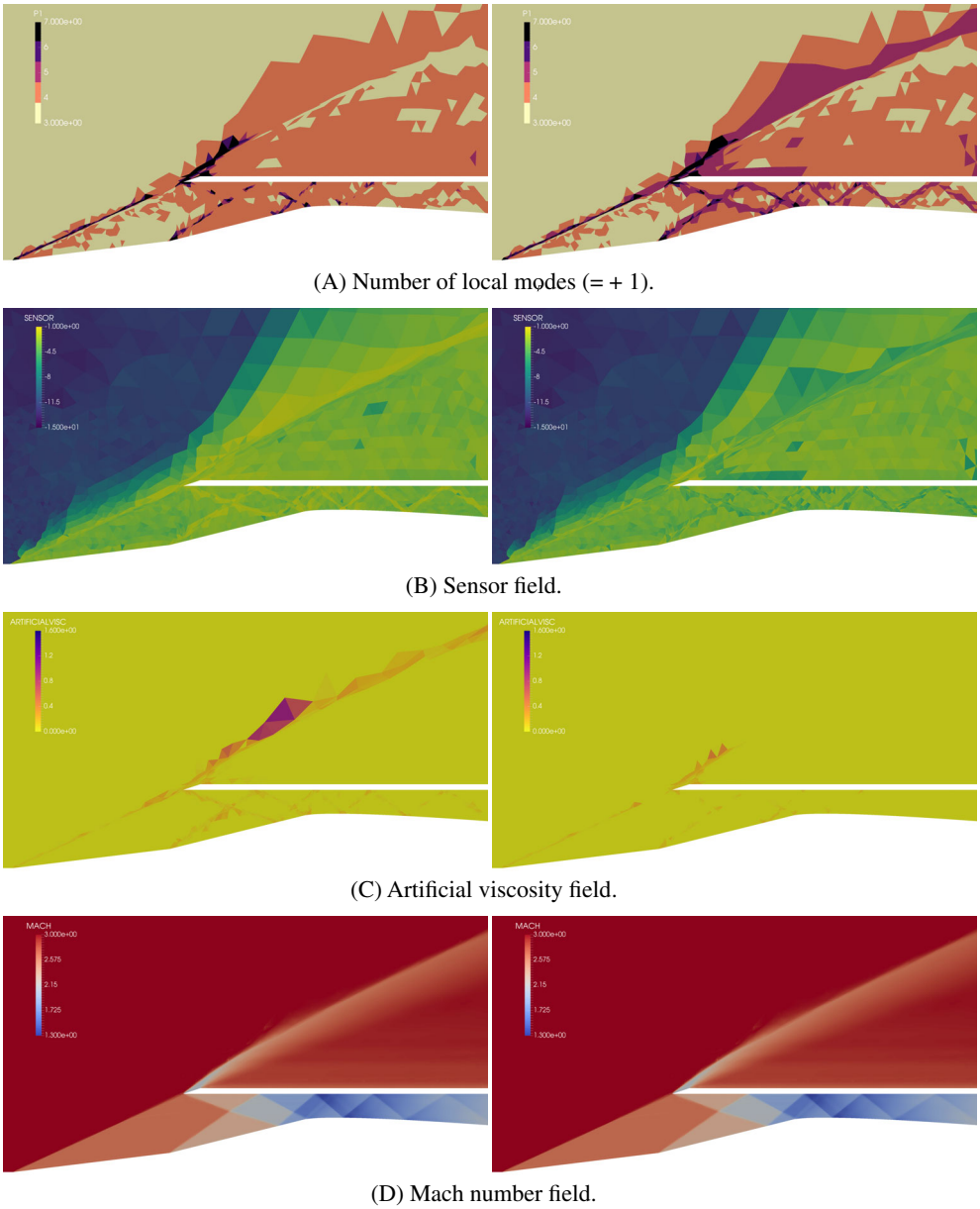
TABLE 2 Number of DOF and CPU time per time step at convergence for the intake. Note that CPU times per time step are reported as run on a 16-core machine, once convergence is reached

Nevertheless we observe that additional resolution in the form of higher local polynomials is found in sensible areas: in the shocks, inside the intake (especially in the throat), and right above the coil. The only very high polynomial orders are obtained in the shocks, whereas smooth regions reach order $p = 3$ at most. The number of DOF for each simulation and state is shown in Table 2. Referring back to Figure 10, we can see that little difference in the solution appears from *rr*-adaptation to *rrp*-adaptation despite the decrease in number of DOF.

8 | CONCLUSIONS

We have presented a novel strategy for adaptive simulations, based on a combination of both *r*- and *p*-adaptation. The proof-of-concept work applied here takes advantage of both strategies in different manners, as appropriate for the simulation of compressible flows containing shocks. We achieve mesh movement required for *r*-adaptation through the use of a variational optimization strategy, using the combination of a local discontinuity sensor and a target element size in order to effectively cluster DOF in the presence of shocks and more sharply simulate their features. At the same time, we apply a *p*-adaptation technique in the rest of the domain in order to benefit from the spectral rate of convergence of high-order discretizations for smooth solutions. The simulation is effectively stabilized through the use of an artificial diffusion term, again using the local discontinuity sensor.

The proposed strategy exhibits a number of benefits from a computational perspective, as seen in the results presented in the previous section, where the traditional NACA 0012 test case and a more challenging supersonic intake have been examined. The main benefit of this dual-adaptive technique is that we are able to significantly reduce the number of DOF required to resolve a given simulation, when compared with a uniformly refined mesh or using solely *r*-adaptation. Table 1



(A) Number of local modes (= + 1).

(B) Sensor field.

(C) Artificial viscosity field.

(D) Mach number field.

FIGURE 11 Comparison of the fields for the supersonic intake in its coarser (left) and finer (right) resolved states during unrestricted *p*-adaptation [Color figure can be viewed at wileyonlinelibrary.com]

shows that, for the various *p*-refinement strategies considered, the error when compared with a very fine solution remains roughly the same, while the simulation requires only 50% of the DOF of the original simulation. This has important consequences from the perspective of computational efficiency, since a significant reduction in the number of DOF will lead to a reduction in runtimes. Likewise, the cost of operations per DOF is reduced as the polynomial order decreases, which offers the opportunity to further reduce computational cost. The *rp*-adaptation technique therefore permits an effective balance to be achieved between the attained error and the simulation expense.

In the context of more general conclusions of our results, we demonstrate that care must be taken when selecting a *p*-adaptation strategy. In particular, the NACA 0012 simulations demonstrate that *p*-coarsening can have important negative effects on the solution for minimal computational gains. Additionally, the supersonic intake exhibits a complex shock pattern. Because of the complexity and strength of the reflecting shocks, we show that multiple *r*-adaptation steps are not only possible but desirable. Despite the lack of nodes to redistribute inside the intake, sufficient mesh deformation is achieved to better capture the various shocks.

Although the overall strategy has been shown to be effective, it is important to emphasize that some of the benefits we highlight in this work can be attributed to our particular implementation of the *r*-adaptation technique. In particular, the use of the variational framework yields several advantages. First, the use of a target element size allows the mesh to deform in an anisotropic manner within restricted regions of the domain. Even when the deformation is substantial, this

still permits a valid mesh to be obtained, as shown in Figure 3. Second, the ability to conform to complex CAD surfaces and curves while permitting nodes to slide across them is clearly important in the context of this work, where shocks arise at or near solid surfaces. This functionality can be difficult to achieve in other mesh deformation techniques, particularly those that require the solution of a partial differential equations (PDE) system of an appropriate solid body model.

Finally, we note that there are a number of clear directions for potential future work in this area. An extension of this method to transient flows, especially with moving shocks, would constitute an interesting application of this *rp*-adaptation strategy. The variational moving mesh framework would be able to track shocks throughout the simulation without the need to generate a new mesh. With preserved mesh connectivities, the system of equations would not need to be rebuilt at each adaptation step. This is especially desirable on large meshes and large simulations based on high-performance computing (HPC) where input-output (I/O) and internode communication can incur significant expense.

The results presented have also unearthed some of the limitations of the approach. In flow simulations with very complex shock patterns, if the original mesh does not contain enough points, increasing the polynomial order on its own will not provide enough DOF to capture these complex shock patterns with sufficient accuracy. Therefore, we posit that incorporating *h*-adaptation will be required and that a combination of the three approaches, namely *hrp*-adaptation, will be required for optimal results.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675008. David Moxey, Spencer J. Sherwin, and Joaquim Peiró acknowledge support from the PRISM project under EPSRC grant EP/R029423/1.

ORCID


Julian Marcon https://orcid.org/0000-0003-3745-7306

REFERENCES

1. de Grazia D, Moxey D, Sherwin SJ, Kravtsova MA, Ruban AI. DNS of a compressible boundary layer flow past an isolated three-dimensional hump in a high-speed subsonic regime. *Phys Rev Fluids*. 2018;3:024101. https://doi.org/10.1103/PhysRevFluids.3.024101.
2. Burbeau A, Sagaut P. A dynamic p-adaptive discontinuous Galerkin method for viscous flow with shocks. *Comput Fluids*. 2005;34(4-5):401-417. https://doi.org/10.1016/j.compfluid.2003.04.002.
3. Li Y, Premasuthan S, Jameson A. Comparison of adaptive h and p refinements for spectral difference methods. Paper presented at: Proceedings of the 40th Fluid Dynamics Conference and Exhibit. American Institute of Aeronautics and Astronautics; 2010; Reston, Virigina.
4. Ekelschot D, Moxey D, Sherwin S, Peiró J. A p-adaptation method for compressible flow problems using a goal-based error indicator. *Comput Struct*. 2016;181:55-69. https://doi.org/10.1016/j.compstruc.2016.03.004.
5. Marcon J, Turner M, Moxey D, Sherwin S, Peiró J. A variational approach to high-order r-adaptation. Paper presented at: Proceedings of the 26th International Meshing Roundtable; 2017; Elsevier, Amsterdam. https://imr.sandia.gov/papers/abstracts/Ma902.html.
6. Mitran S. A Comparison of Adaptive Mesh Refinement Approaches for Large Eddy Simulation. *Proceedings of the Third AFOSR International Conference on DNS/LES*. Seattle; Fort Belvoir, VA: Washington University; 2001. https://apps.dtic.mil/dtic/tr/fulltext/u2/p013657.pdf.
7. Oden J, Wu W, Legat V. An hp-adaptive strategy for finite element approximations of the Navier-Stokes equations. *Int J Numer Methods Fluids*. 1995;20:831-851. https://doi.org/10.1002/fld.1650200810.
8. Persson P, Peraire J. Sub-cell shock capturing for discontinuous Galerkin method. Paper presented at: Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit. American Institute of Aeronautics and Astronautics; 2006; Reno, Nevada, AIAA Paper 2006-112. https://doi.org/10.2514/6.2006-112.
9. Fidkowski K, Darmofal D. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA J*. 2011;49:673-694. https://doi.org/10.2514/1.J050073.
10. Yano M. An Optimization Framework for Adaptive Higher-Order Discretizations of Partial Differential Equations on Anisotropic Simplex Meshes [PhD thesis]. Massachusetts Institute of Technology, Cambridge, MA; 2012. https://dspace.mit.edu/handle/1721.1/76090.
11. Naddei F, de la Llave PM, Couaillier V, Coquel F. A comparison of refinement indicators for p-adaptive simulations of steady and unsteady flows using discontinuous Galerkin methods. *J Comput Phys*. 2019;376:508-533. https://doi.org/10.1016/j.jcp.2018.09.045.
12. Offermans N, Peplinski A, Marin O, Schlatter P. Adaptive mesh refinement for steady flows in Nek5000. *Comput Fluids*. 2020;197:104352.
13. Jacobs CT, Zauner M, De Tullio N, Jammy SP, Lusher DJ, Sandham ND. An error indicator for finite difference methods using spectral techniques with application to aerofoil simulation. *Comput Fluids*. 2018;168:67-72.
14. Moxey D, Cantwell C, Mengaldo G, et al. Towards p-adaptive spectral/hp element methods for modelling industrial flows. *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM*. Vol 2017. Cham: Springer; 2016:63-79.

15. Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J Comput Phys*. 1996;131:267-279. https://doi.org/10.1006/jcph.1996.5572.
16. Cantwell CD, Moxey D, Comerford A, et al. Nektar++: an open-source spectral/hp element framework. *Comput Phys Commun*. 2015;192:205-219. https://doi.org/10.1016/j.cpc.2015.02.008.
17. Moxey D, Cantwell CD, Bao Y, et al. Nektar++: enhancing the capability and application of high-fidelity spectral/hp element methods. *Comput Phys Commun*. 2020;249:107110.
18. Cockburn B, Shu C. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J Numer Anal*. 1998;35(6):2440-2463. https://doi.org/10.1137/S0036142997316712.
19. Barter GE, Darmofal DL. Shock capturing with PDE-based artificial viscosity for DGFEM: Part I formulation. *J Comput Phys*. 2010;229(5):1810-1827. https://doi.org/10.1016/j.jcp.2009.11.010.
20. Turner M, Peiró J, Moxey D. Curvilinear mesh generation using a variational framework. *Comput-Aid Des*. 2017;103:73-91. https://doi.org/10.1016/j.cad.2017.10.004.
21. Klöckner A, Warburton T, Hesthaven JS. Viscous shock capturing in a time-explicit discontinuous Galerkin method. *Math Modell Natural Phenomena*. 2011;6(3):57-83.
22. Vassberg J, Jameson A. In pursuit of grid convergence for two-dimensional Euler solutions. *J Aircraft*. 2010;47(4):1152-1166. https://doi.org/10.2514/1.46737.
23. Turner M, Moxey D, Sherwin S, Peiró J. Automatic generation of 3D unstructured high-order curvilinear meshes. Paper presented at: Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering. ECCOMAS; 2016; Crete Island, Greece. https://doi.org/10.7712/100016.1825.8410.
24. Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Berlin, Heidelberg/Germany: Springer; 2009. https://link.springer.com/book/10.1007/b79761.
25. Open Cascade SAS Open Cascade. Software; 2019. https://www.opencascade.com/
26. Anderson WE, Wong ND. Experimental Investigation of a Large-Scale, Two-Dimensional, Mixed-Compression Inlet System - Performance at Design Conditions, M=3.0. *Technical Memorandum NASA TM X-2016*. Moffet Field, CA: NASA Ames Research Center; 1970. https://ntrs.nasa.gov/citations/19700018205.
27. Jain M, Mittal S. Euler flow in a supersonic mixed-compression inlet. *Int J Numer Methods Fluids*. 2006;50(12):1405-1423. https://doi.org/10.1002/fld.1109.

How to cite this article: Marcon J, Castiglioni G, Moxey D, Sherwin SJ, Peiró J. *rp*-adaptation for compressible flows. *Int J Numer Methods Eng*. 2020;1–21. https://doi.org/10.1002/nme.6529